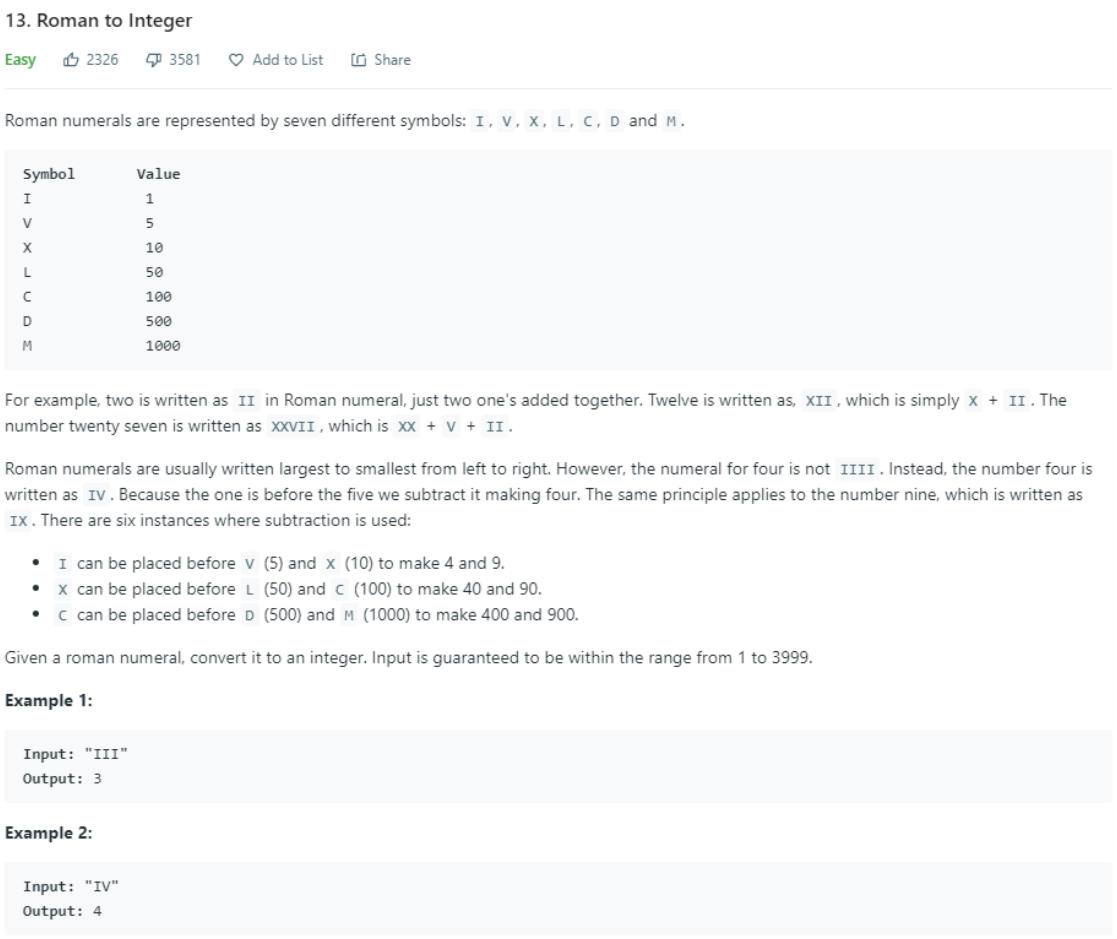


13. Roman to Integer

Roman numerals are represented by seven different symbols: `I`, `V`, `X`, `L`, `C`, `D` and `M`.

```
Symbol       Value
I             1
V             5
X             10
L             50
C             100
D             500
M             1000
```

For example, two is written as `II` in Roman numeral, just two one's added together. Twelve is written as, `XII`, which is simply `X` + `II`. The number twenty seven is written as `XXVII`, which is `XX` + `V` + `II`.

Roman numerals are usually written largest to smallest from left to right. However, the numeral for four is not `IIII`. Instead, the number four is written as `IV`. Because the one is before the five we subtract it making four. The same principle applies to the number nine, which is written as `IX`. There are six instances where subtraction is used:

- `I` can be placed before `V` (5) and `X` (10) to make 4 and 9.
- `X` can be placed before `L` (50) and `C` (100) to make 40 and 90.
- `C` can be placed before `D` (500) and `M` (1000) to make 400 and 900.

Given a roman numeral, convert it to an integer. Input is guaranteed to be within the range from 1 to 3999.

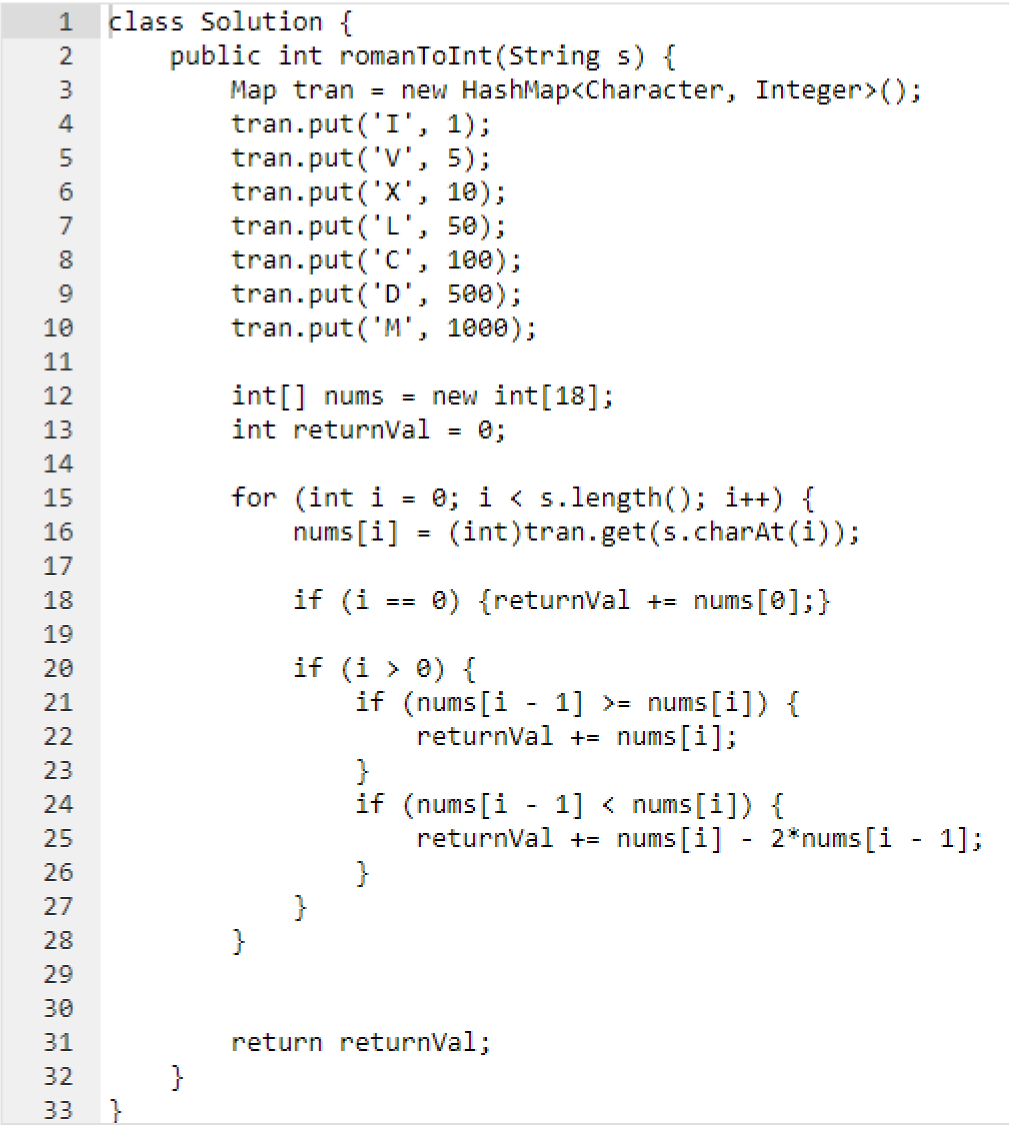
Example 1:

```
Input: "III"
Output: 3
```

Example 2:

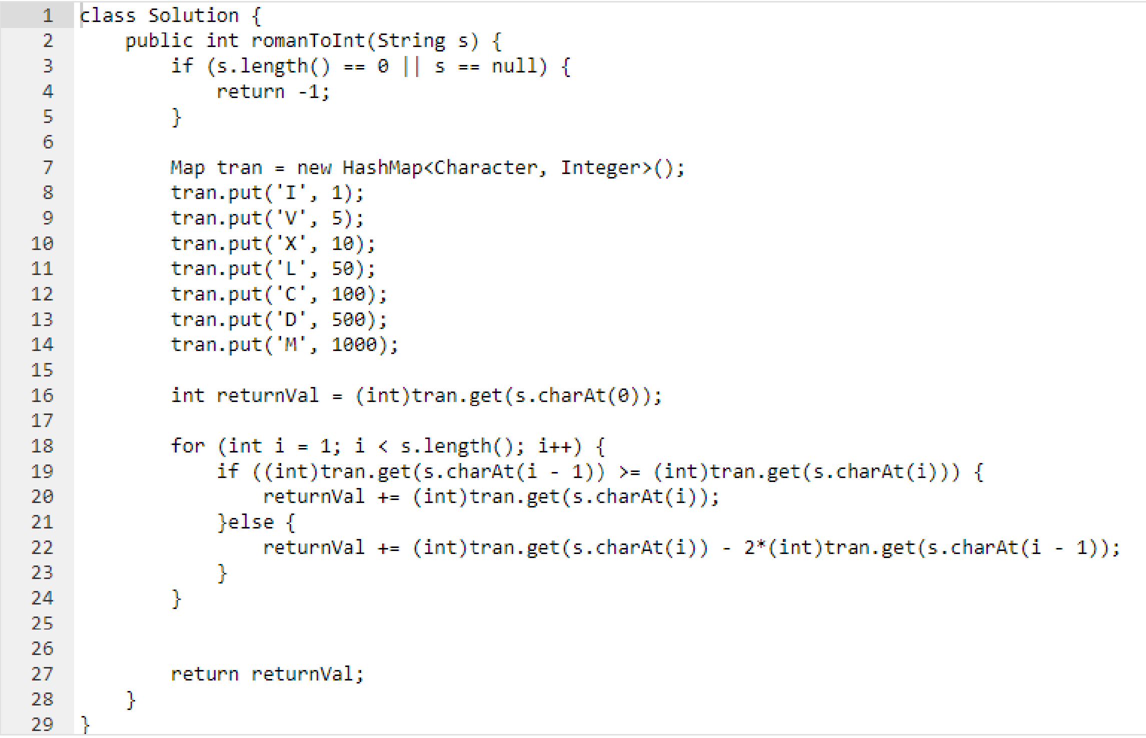
```
Input: "IV"
Output: 4
```

Sub 1



```
class Solution {
    public int romanToInt(String s) {
        Map tran = new HashMap<Character, Integer>();
        tran.put('I', 1);
        tran.put('V', 5);
        tran.put('X', 10);
        tran.put('L', 50);
        tran.put('C', 100);
        tran.put('D', 500);
        tran.put('M', 1000);

        int[] nums = new int[18];
        int returnVal = 0;

        for (int i = 0; i < s.length(); i++) {
            nums[i] = (int)tran.get(s.charAt(i));

            if (i == 0) {returnVal += nums[0];}

            if (i > 0) {
                if (nums[i - 1] >= nums[i]) {
                    returnVal += nums[i];
                }
                if (nums[i - 1] < nums[i]) {
                    returnVal += nums[i] - 2*nums[i - 1];
                }
            }
        }


        return returnVal;
    }
}
```

Sub 2



```
class Solution {
    public int romanToInt(String s) {
        if (s.length() == 0 || s == null) {
            return -1;
        }

        Map tran = new HashMap<Character, Integer>();
        tran.put('I', 1);
        tran.put('V', 5);
        tran.put('X', 10);
        tran.put('L', 50);
        tran.put('C', 100);
        tran.put('D', 500);
        tran.put('M', 1000);

        int returnVal = (int)tran.get(s.charAt(0));

        for (int i = 1; i < s.length(); i++) {
            if ((int)tran.get(s.charAt(i - 1)) >= (int)tran.get(s.charAt(i))) {
                returnVal += (int)tran.get(s.charAt(i));
            }else {
                returnVal += (int)tran.get(s.charAt(i)) - 2*(int)tran.get(s.charAt(i - 1));
            }
        }


        return returnVal;
    }
}
```